Towards a Firmware Update Standard

Mallik Bulusu  
UEFI Dev. Lead  
Microsoft Corporation  
mallikb@microsoft.com

Vincent Zimmer  
Sr. Principal Engineer  
Intel Corporation  
vincent.zimmer@intel.com
What Did We Say in 2015?

Cloud scale offers unique challenges to development of firmware.

Factors affecting FW updates –
- Different Types
- Deployment
- Conformance
- Availability
- Recovery

We can’t avoid FW updates –
- Bug Fixes
- Performance Improvements
- Security

Firmware Update Challenges
- Components from multiples vendors
- Delivering firmware
- Different types of devices
- Recovery from failures
- Node equivalence across datacenter
- Security, security, security…….
# FW Update Scenarios in Cloud

<table>
<thead>
<tr>
<th>Mode</th>
<th>Advantages</th>
<th>Opens</th>
</tr>
</thead>
<tbody>
<tr>
<td>UEFI</td>
<td>API / Envelope definition; System/Device Firmware story;</td>
<td>System Resets;</td>
</tr>
<tr>
<td>SMM</td>
<td>Silent</td>
<td>Device Firmware story. Few updates take affect after reset.</td>
</tr>
<tr>
<td>OS Driver</td>
<td>Silent</td>
<td>Standardization (?)</td>
</tr>
<tr>
<td>OOB</td>
<td>Controlled Environment</td>
<td>Arbitration with host context; Bandwidth</td>
</tr>
</tbody>
</table>

**NOMENCLATURE:**

**System Firmware:** Elements that **are required** for system boot, for e.g. BIOS, ME, BMC.

**Device Firmware:** Elements that **are not required** for system boot, for e.g. OPROMs, etc.
What is a Capsule?

- **API**
  - System or Device initialization
  - Carries interfaces to interact with Device
  - Presents interfaces to Read, Modify, Verify Firmware

- **Envelope**
  - Parsing the blob
  - Integrity checks
  - Validity checks
  - Update mode independent
  
  Prevent rogue FW updates
Provisions in UEFI

- Publish metadata (ESRT)
  - Firmware Resource Description
  - Published by BIOS
  - Firmware update status
- Capsules on Disk
- Smart capsules (FMP)
  - Published by driver
  - Capsule comprises of header & body
  - Capsule body comprises
    - EFI Firmware Management Capsule Header
    - Optional Drivers
    - Payloads
    - Updates handled during pre-boot

Can we extend these provisions to other Firmware Update scenarios?
Envelope is More Universal

- The capsules itself has a header and body
- Capsule can be firmware volume with encapsulation sections
- Sections can include compression, signing, encryption
- Some well-known section types in the UEFI PI specification
- Can be vendor specific
How to Use the Envelope?

- **In-band** – EFI UpdateCapsule or Capsule-on-Disk
- **Out-of-Band** – via Service Processor
- **OS** – Linux FW API, MS .cab etc.

Capsules are GUID based
- Some well known GUIDs
- Others can be
  - Vendor specific;
  - Standards group specific (think OCP?)
Security

- Envelope to also include signing.
- Already have UEFI Secure Boot for UEFI executables (apps & drivers)
- Have UEFI envelope possibilities.
  - Security Version Numbers (SVN’s)
  - golden images
  - roll-back concerns
Questions?

- Security (see before)
- Should in band (e.g., UpdateCaspule) and out of band (OOB) be harmonized?
- Same binary to OS driver for device?
- How to get inventory for node equivalence?
- Share tools?
- Namespace of GUID’s for OCP style gear?

What type of information that could appear in an OCP spec on ‘updates?’ What can land in ’16 given aforementioned ecosystem readiness?
More information

http://www.opencompute.org – OCP specs

http://www.uefi.org – UEFI, ACPI, Shell, PI Specifications

http://www.Tianocore.org – open source UEFI

http://firmware.intel.com – white papers, training

https://www.dmtf.org/sites/default/files/UEFI-DMTFWorkReg1_2_v2.pdf – DMTM & UEFI


https://firmware.intel.com/blog/uefi-and-cloud – UEFI & Cloud discussion at UEFI Plugfest


https://blog.uncooperative.org/blog/2015/09/16/an-update-on-firmware-updates/ – Linux Firmware Updates